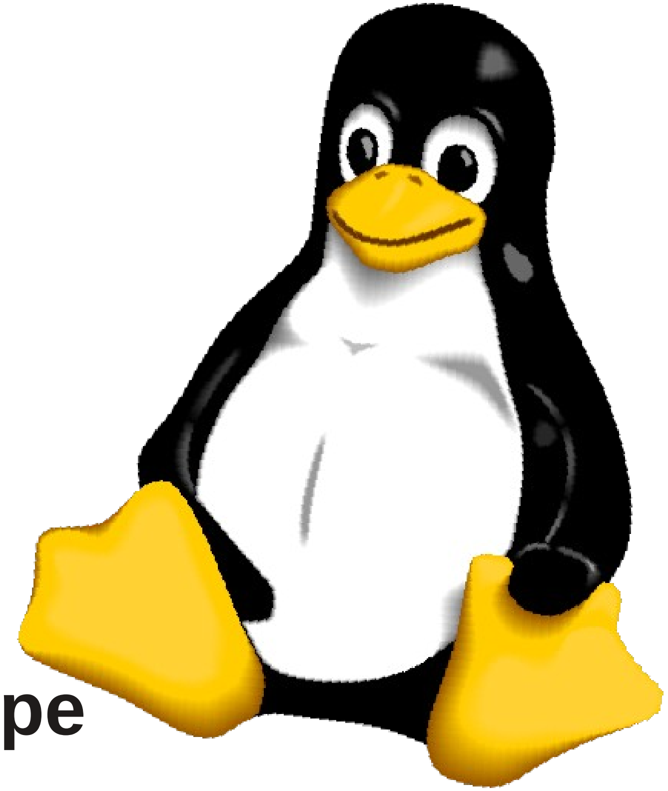
Wireless Networking with IEEE 802.15.4 and 6LoWPAN

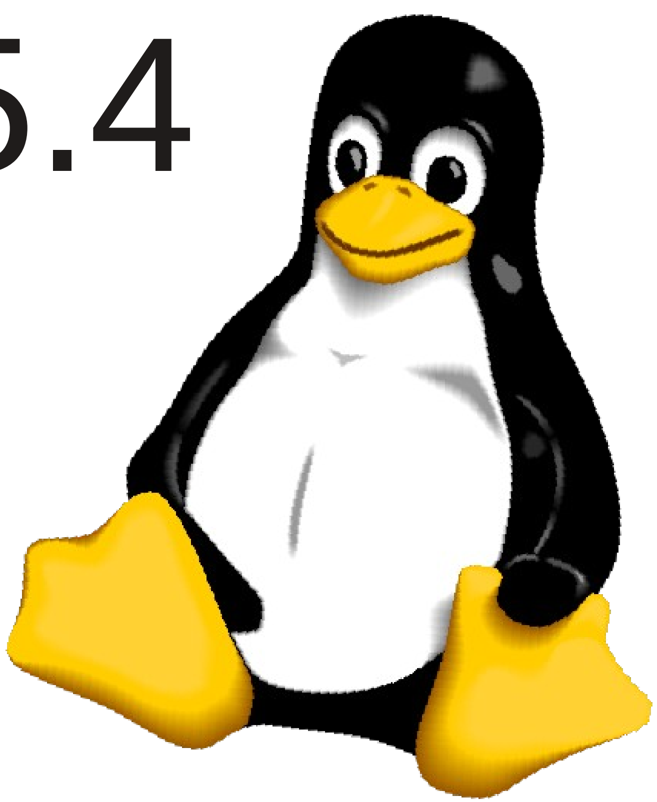


Alan Ott

Embedded Linux Conference – Europe

November 5, 2012

IEEE 802.15.4



IEEE 802.15.4

- IEEE 802.15.4 is a standard for **low-power**, **low data rate** wireless communication between small devices.
- Forms the basis for Low Rate, Wireless Personal Area Networks (LR-WPANs)
  - Low transmitter power
  - Small MTU
  - Low power consumption
  - Low cost

IEEE 802.15.4

“I've heard of that; you mean ZigBee.”

- 802.15.4 is not the same thing as ZigBee.
- 802.15.4 is a MAC and PHY layer protocol (OSI layers 1 and 2).
- ZigBee is a Network Layer (OSI layer 3) protocol which sits on top of IEEE 802.15.4.
  - There are several layer 3 protocols which can make use of 802.15.4

A Word About ZigBee

- ZigBee is a trademark of the **ZigBee Alliance,** the group which creates and maintains the standard.
- The ZigBee standard is available for no charge for **non-commerical** purposes only.
- A **paid membership** in the ZigBee Alliance is required in order to produce products which use ZigBee.

A Word About ZigBee

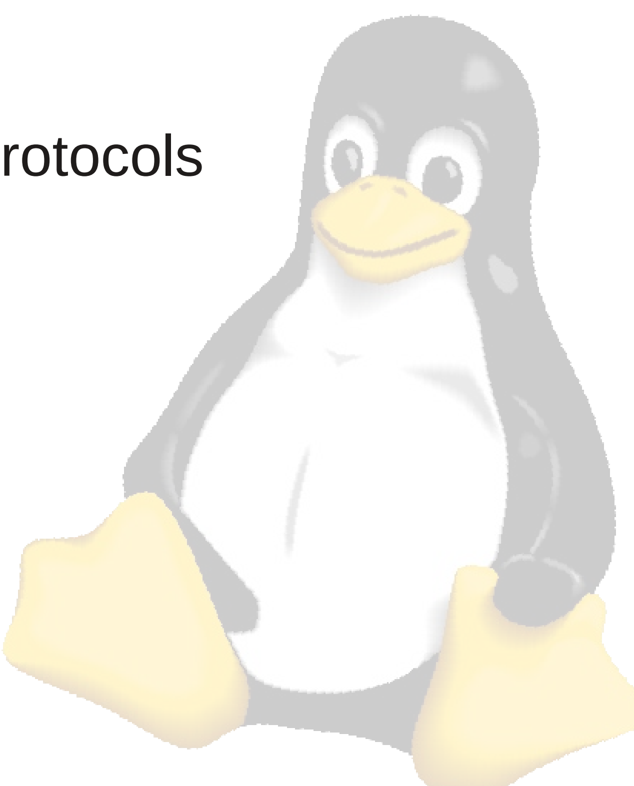
- ZigBee's license **conflicts** with the GPL and other Free Software licenses.
- Until the ZigBee Alliance changes their license, there will **likely not ever be** an implementation of ZigBee in the Linux kernel.

A Word about Zigbee

- Zigbee IP Stack
  - Not to be confused with 802.15.4 and 6LoWPAN
    - Zigbee Alliance Protocol which is based on Zigbee and 6LoWPAN.

IEEE 802.15.4

- Higher-level Protocols which make use of 802.15.4:
  - **Zigbee**
    - Zigbee Alliance's mesh networking protocol
  - **MiWi Mesh** and **MiWi P2P**
    - Microchip's proprietary mesh and P2P protocols
  - **6LoWPAN**
    - IPv6 over 802.15.4
  - **WirelessHART**
    - Industrial Automation
  - **ISA100.11a**
    - Manufacturing, Control, Automation



IEEE 802.15.4

- Specifications
  - Operates on several bands:
    - **2.4 GHz** ISM band
      - (Q-QPSK at 250 kb/s)
    - **915 MHz**
      - (BPSK at 40 kb/s, ASK at 250 kb/s, Q-QPSK at 250 kb/s)
    - **868 MHz**
      - (BPSK at 20 kb/s, ASK at 250 kb/s, Q-QPSK at 100 kb/s)

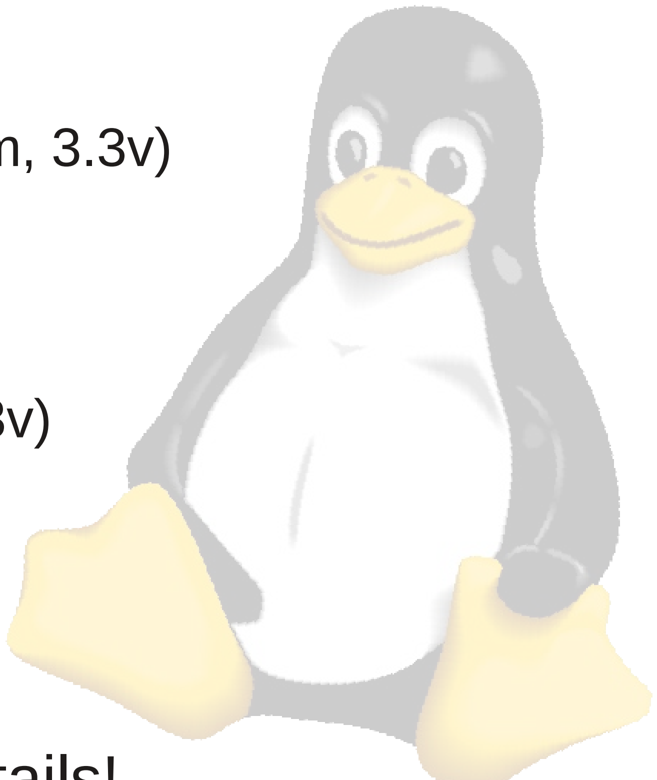
IEEE 802.15.4

- Specifications
  - Output Power
    - **2.4 GHz**
      - 20 dBm (100 mW) (US/Europe)
    - **915 MHz**
      - > 10 dBm
    - **868 MHz**
      - 30 dBm (1 W US)

    - Check your local regulations.
      These numbers are not legal advice!

IEEE 802.15.4

- Specifications
  - Power Draw
    - – Microchip **MRF24J40MA** (2.4GHz, 0 dBm, 3.3v)
      - 19 mA RX (typ)
      - 23 mA TX (typ)
    - – Texas Instruments **CC2420** (2.4GHz, 0 dBm, 3.3v)
      - 18.8 mA RX
      - 17.4 mA TX
      - 426 uA Idle
    - – Freescale **MC13202** (2.4GHz, 3.6 dBm, 3.3v)
      - 37 mA RX
      - 30 mA TX
      - 500 uA Idle



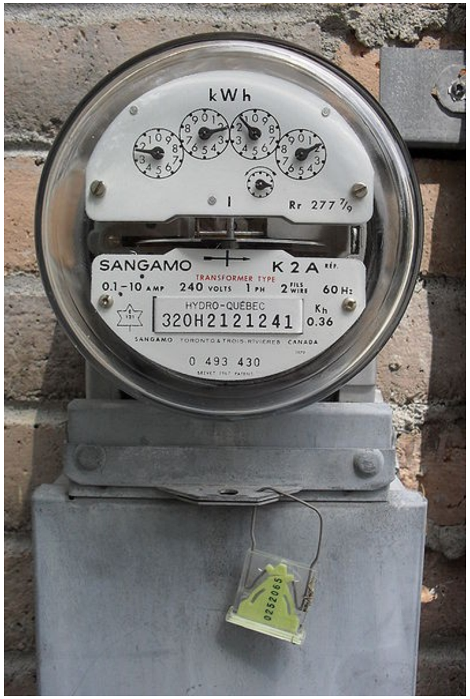
➔ Consult datasheets for details!

IEEE 802.15.4

- Specifications
  - Data Rate
    - Up to **250 kb/s** depending on band and mode
    - Higher if proprietary modes are used
      - (MRF24J40 can do 625 kb/s in Turbo mode)
  - MTU
    - **127** Bytes per frame (including headers)
    - 802.15.4g is likely to bring a 2047-byte MTU
      - This will of course require different hardware

IEEE 802.15.4

- Uses of 802.15.4
  - Industrial control and monitoring
  - Wireless sensor networks
  - Intelligent agriculture
  - Security systems
  - Smart Grid



Images from Wikipedia

Embedded Linux Conference Europe

IEEE 802.15.4

- Types of Devices
  - **Full Function Device** (FFD)
    - Can talk to all types of devices
    - Supports full protocol
  - **Reduced Function Device** (RFD)
    - Can only talk to an FFD
    - Lower power consumption
    - Minimal CPU/RAM required

IEEE 802.15.4

- PANs
  - Devices are segregated into Personal Area Networks (PAN)
    - Multiple PANs can operate on a single channel.
    - Each PAN has a **PAN Identifier**
    - Devices can communicate **between PANs** (inter-PAN) or within their own PAN (intra-PAN).

IEEE 802.15.4

- PAN Identifier
  - **16-bit** number
  - Does not need assignment from a central authority
    - No large sums of money involved like with USB or Zigbee
  - PAN ID can be **pre-determined** or **scanned for** at coordinator start-up time.
    - Can scan for a fixed PAN ID on each channel
    - Can scan for multiple PAN ID's on a single channel
  - Frames can be sent inter-PAN
  - Broadcast PAN ID is `0xffff` (all PANs)

IEEE 802.15.4

- Addressing
  - Each device has two addresses
    - **Long Address**
      - **64-bit** globally unique device ID
    - **Short Address**
      - **16-bit** PAN-specific address
      - Assigned by the PAN coordinator at association time
  - Broadcast address
    - Addresses all Nodes in a PAN
    - Short Address: `0xffff`

➔ Short and long addresses may be mixed in a MAC header.

IEEE 802.15.4

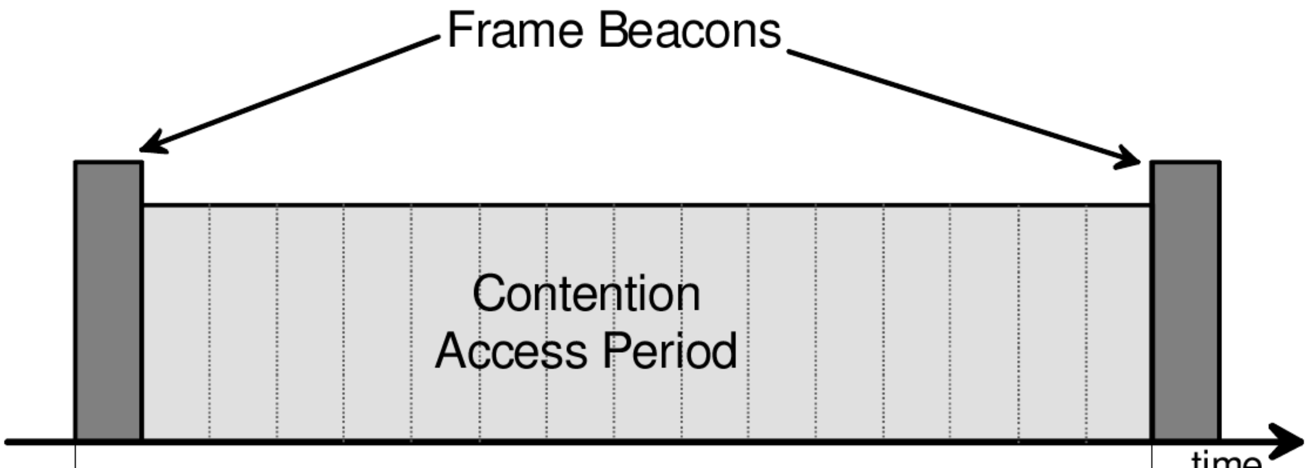
- Coordinator
  - Each PAN has a **PAN coordinator**
    - Full-function device (FFD)
    - Processes requests to join/leave the network
    - Assigns short addresses to devices
      - Short addresses are optional

IEEE 802.15.4

- Beacon-Enabled Networks
  - IEEE 802.15.4 networks can optionally be beacon-enabled.
    - The PAN Coordinator sends a beacon frame to synchronize and delineate **Superframes**.
    - Access to the channel is **slotted**.
    - Superframes can contain **Guaranteed Time Slots** (GTS), each of which can be assigned to a specific device, preventing media access contention.
    - Beacon-enabled networks enable devices to consume **less power**, because the receivers can be switched off during parts of the superframe.

IEEE 802.15.4

- Beacon-Enabled Network Superframe

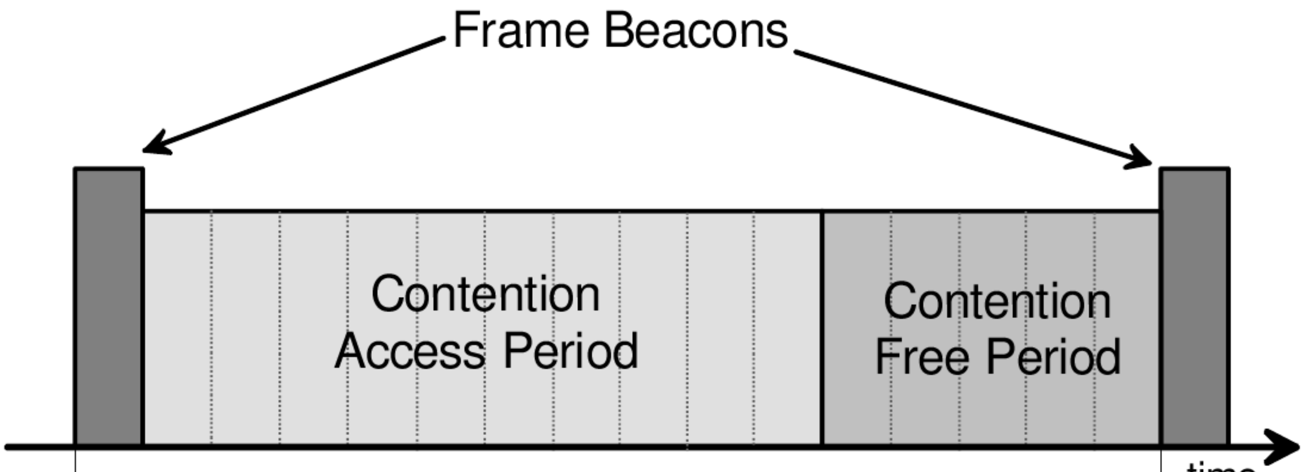


Source IEEE 802.15.4-2003 Spec

- Frames must be sent in one of the slots.
  - 16 slots total, one of which contains the beacon frame.

IEEE 802.15.4

- Beacon-Enabled Network Superframe with Guaranteed Time Slots (GTS)



Source IEEE 802.15.4-2003 Spec

- Slots in the Contention-Free Period are each reserved for individual devices.

IEEE 802.15.4

- Beaconless networks
  - No beacon frames transmitted by the coordinator
  - Receivers must be listening all the time
  - Full-time **contention-access**
  - **Unslotted**
  - Uses **more battery**, but easier to configure

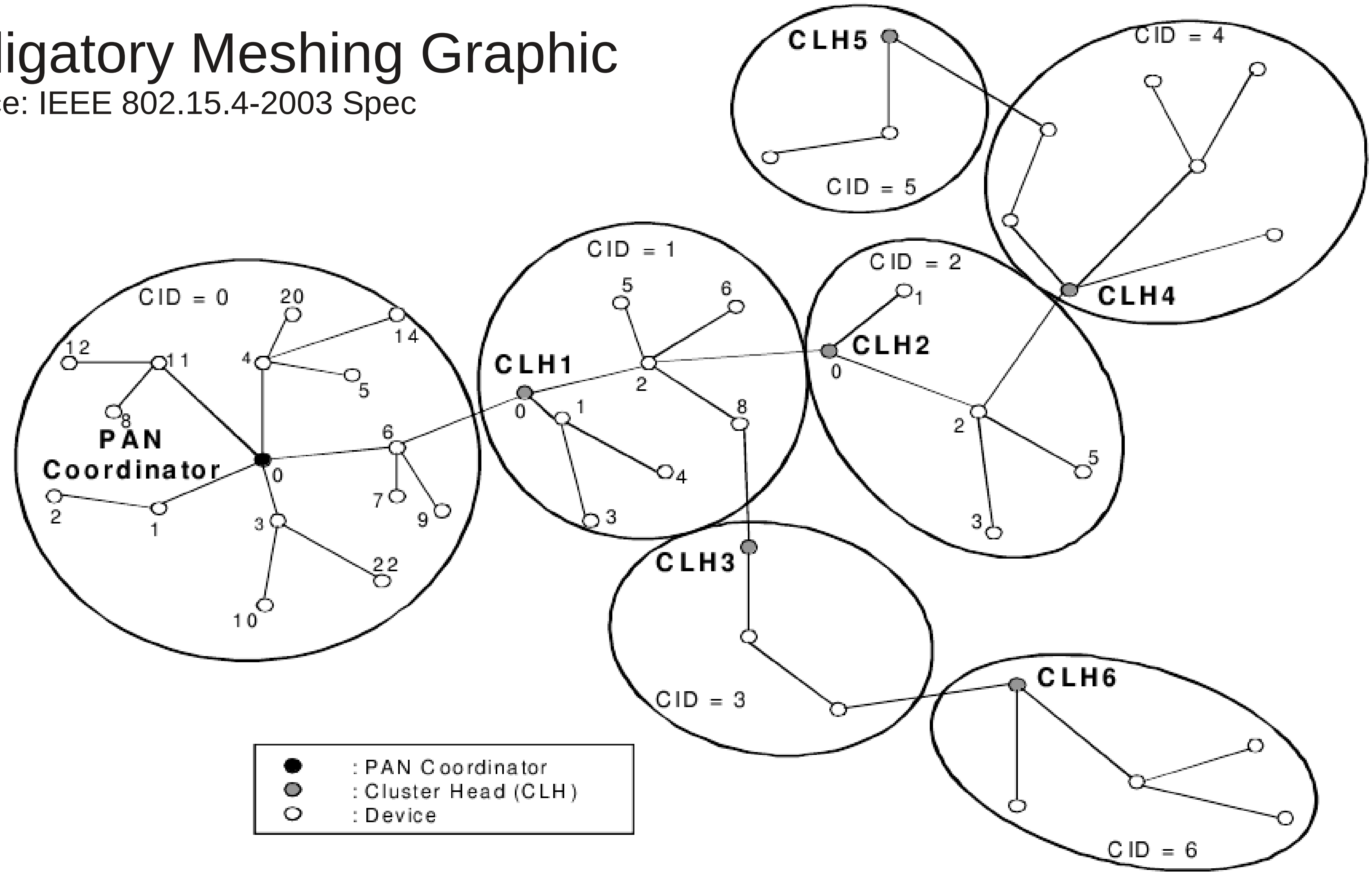
IEEE 802.15.4

- Meshing
  - **Meshing** is the ability to route messages through multiple hops on the network between source and destination.
  - While 802.15.4 is designed with meshing in mind, it is not part of the 802.15.4 standard, and left to the **network layer**.
    - ZigBee and MiWi support meshing

IEEE 802.15.4

Obligatory Meshing Graphic

Source: IEEE 802.15.4-2003 Spec



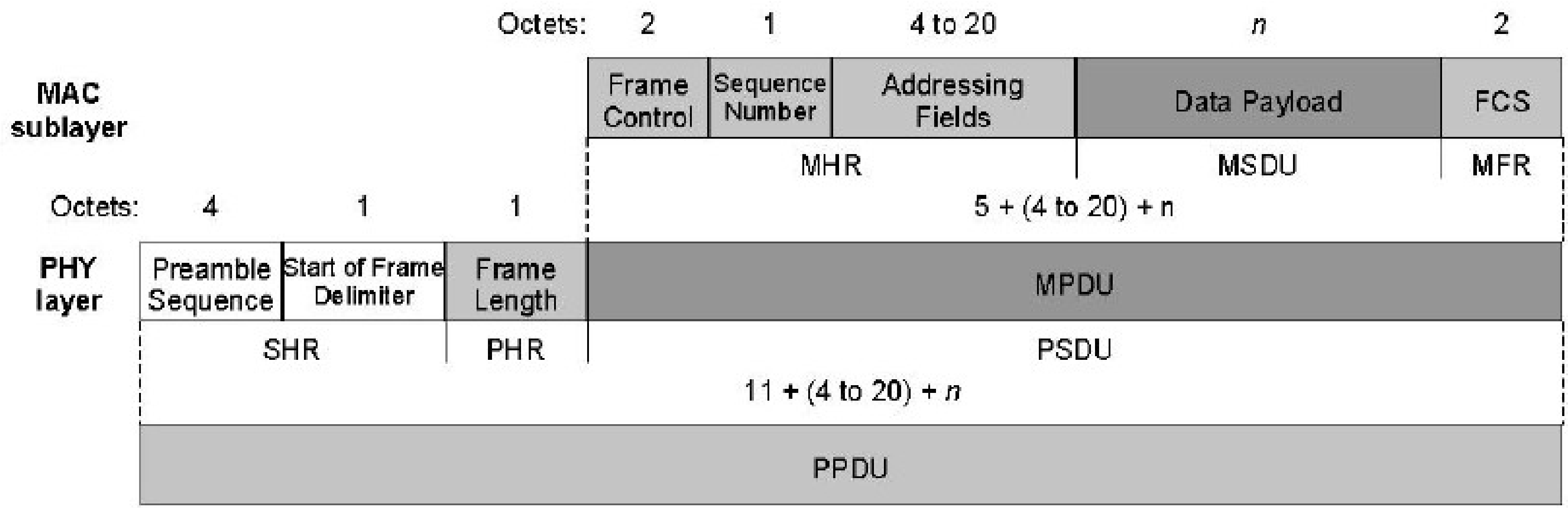
IEEE 802.15.4

- Frame Types
  - Four Types of frame
    - **Beacon** Frame
      - Sent by Coordinator to set up the Superframe structure.
    - **Data** Frame
      - Transfers application data.
    - **Acknowledgement** Frame
      - Provide confirmation of reception
    - **MAC Command** Frame
      - MAC-layer network management
        - Associate, Disassociate, Beacon request, GTS request

IEEE 802.15.4

- Data Frame Format

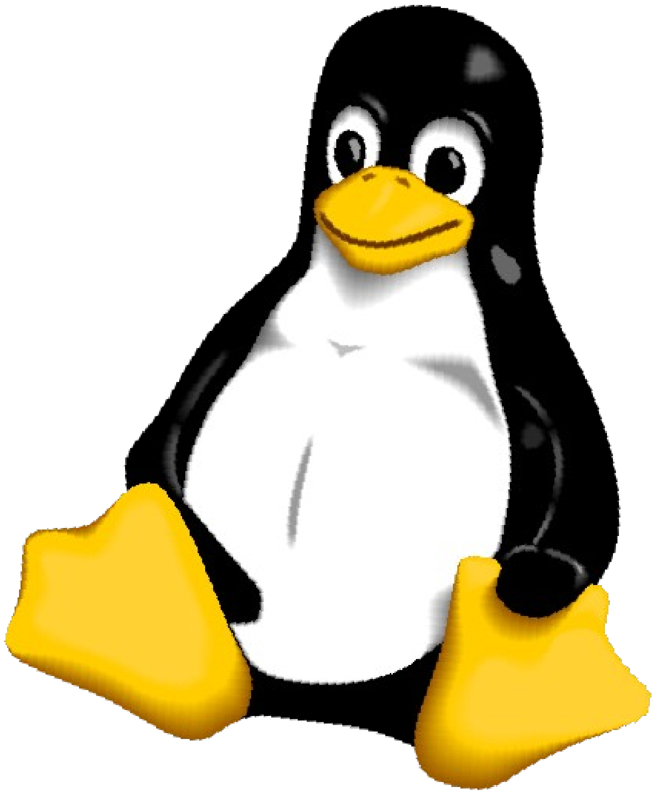
Source: IEEE 802.15.4-2003 Spec



IEEE 802.15.4

- Security
  - AES encryption
    - Several modes of **encryption** with increasing levels of complexity and security are available.
    - Using lower security when appropriate will reduce computational complexity and **save battery life**.
    - **Pre-shared key**, symmetric cryptography

6LoWPAN



6LoWPAN

- Overview
  - It is desirable to use IP to communicate with small devices.
    - Widely deployed
    - IPv6's addressing space is large, allowing even small devices to have a real-world routable IPv6 address.
  - MTU issues:
    - IPv6 has an MTU requirement of **1280** bytes.
    - 802.15.4 has an MTU of **127** bytes.

6LoWPAN

- Overview
  - Other IPv6 issues
    - The header overhead is large
      - 802.15.4 maximum frame overhead of 25 bytes
      - Link-layer security can be as high as 21 bytes
        - ***This leaves 81 bytes left***
      - 40-byte IP header
      - 8-byte UDP header
        - ***33 bytes remaining for actual data***
    - This is clearly less than desirable

6LoWPAN

- Overview
  - Need a way to wedge IPv6 into 802.15.4
  - The solution: **6LoWPAN** (RFC 4944 and RFC 6282)
    - Packet fragmentation **below** the Network Layer
    - Header Compression
      - Compress IP addresses when they can be derived from other headers, such as the 802.15.4 MAC header.
        - Compress Prefix for link-local (fe80::)
        - Elide address completely when it can be fully derived from the link-layer address.
      - Compress common headers:
        - TCP, UDP, ICMP

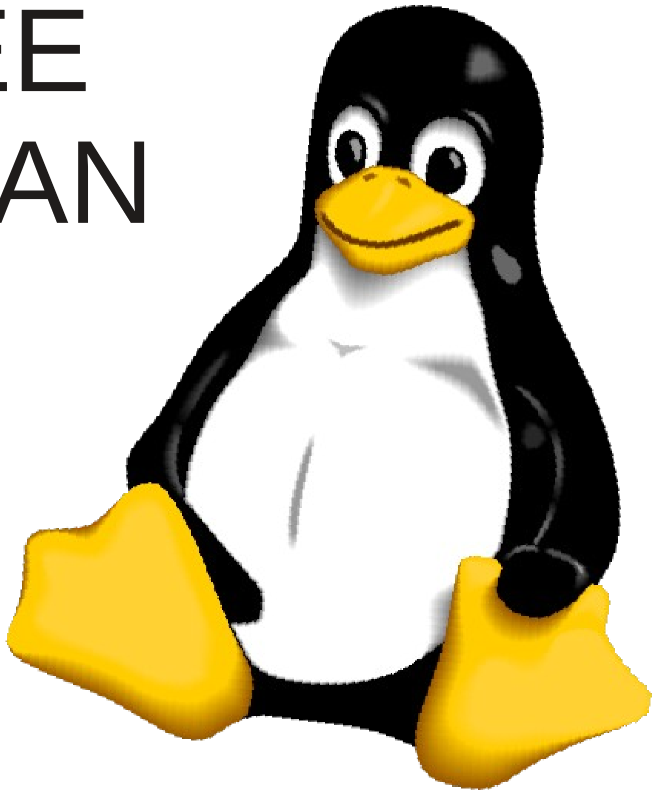
6LoWPAN

- Overview
  - Meshing
    - 6LoWPAN has a **Mesh Address Header**, to support routing of packets in a mesh network, but leaves the details of routing to the **link layer**.
    - Remember that 802.15.4 leaves mesh routing the **network layer**.
    - Result? Good luck with meshing.

6LoWPAN

- Implications
  - Using 6LoWPAN and IPv6, every small device can have a routable IP.
    - This makes administration much easier
    - It also makes security more important
  - Standard tools can be used to administer small devices.
    - Web-based interfaces
    - ssh, telnet, FTP, etc.

Linux Support for IEEE 802.15.4 and 6LoWPAN



Embedded Linux Conference Europe

Support in Linux

- Projects
  - There are currently two kernel trees, and two project websites.
    - **Linux-Zigbee** project
      - http://linux-zigbee.sourceforge.net
    - **Linux-wsn** project
      - http://code.google.com/p/linux-wsn/
  - There is work being done to fix this up

Support in Linux

- **Linux-Zigbee** Project
  - Started by engineers at Siemens
  - Originally intended to provide an in-kernel Zigbee implementation
    - Once licensing incompatibilities were discovered, this goal shifted to implementing 802.15.4 and 6LoWPAN.
  - Status
    - Project kernel (based on 3.3-rc5) has working implementation of 802.15.4 and some 6LoWPAN.
    - Key players have since been re-assigned
    - Kernel hasn't been updated in 6 months

Support in Linux

- **Linux-Zigbee** Project
  - Userspace tools
    - `iz` – network device **configuration** tool
    - `izcoordinator` – **PAN coordinator** implementation
    - `izchat` – simple raw 802.15.4 chat program for testing.
  - Drivers
    - Atmel AT86RF230
    - Texas Instruments CC2420
    - Analog Devices ADF7272
    - Redwire Econotag (uses serial.c)

Support in Linux

- **Linux-wsn** Project
  - After re-assignment of Siemens engineers, **Alexander Smirnov** started getting the work from Linux-zigbee into the mainline kernel.
  - Current **mainline Linux kernel** now contains the most up-to-date implementation.
  - New patches go through Dave Miller's **net-next** tree.

Support in Linux

- **Linux-wsn** Project
  - Current Support:
    - Same userspace tools as Linux-zigbee
    - 802.15.4 Raw sockets
    - 6LoWPAN
  - Drivers
    - Atmel AT86RF230
    - Microchip MRF24J40
    - Redwire Econotag (currently out-of-tree)

Support in Linux

- Limitations
  - 802.15.4 **TODO** list
    - Beacon-enabled networks (with and without GTS)
    - Security
    - Association / disassociation
    - Scanning
    - Acknowledgement
    - More Device drivers
    - Likely much much more

Support in Linux

- Limitations
  - 6LoWPAN Current Limitations
    - Not all address compression types are supported.
      - Communication between Linux nodes is OK
      - Communication between Linux and other OS's is not
    - Uncompressed headers not supported
    - Some header types are not supported

Support in Linux

- Supported Features
  - Don't be put off, there's a lot of stuff that ***does*** work!
  - IPv6 communication works between Linux devices
    - ssh, ping6, etc.
  - Packet capturing with **tcpdump** and **Wireshark**.

Support in Linux

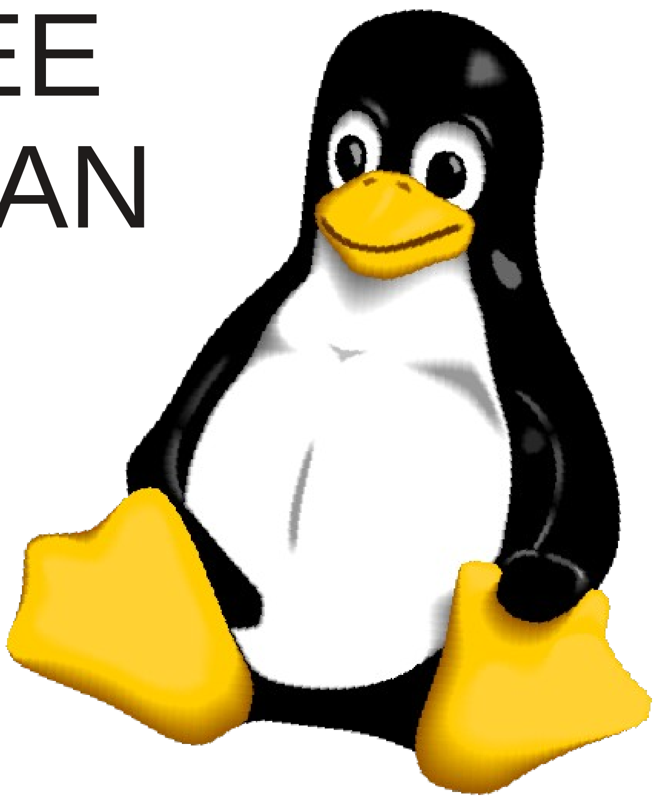
Configuring a device:

```
iz listphy # show all wpan-phy physical devices
iz add wpan-phy0 # create wpan0 attached to wpan-phy0
ip link set wpan0 address a0:a0:a0:a0:a0:a0:a0:a0
ifconfig wpan0 up

# Set the PAN ID, channel and short address.
# This is a temporary hack. iz assoc eventually be used.
export PID_FILE=/var/run/izpid
izcoordinator -i wpan0 -d 1 -s 2 -p 777 -c 11 -l lease &
sleep 1

# Create a 6LoWPAN link and set its hardware address
ip link add link wpan0 name lowpan0 type lowpan
ip link set lowpan0 address a0:0:0:0:0:0:0:2
ifconfig lowpan0 up
```

Other Support for IEEE 802.15.4 and 6LoWPAN



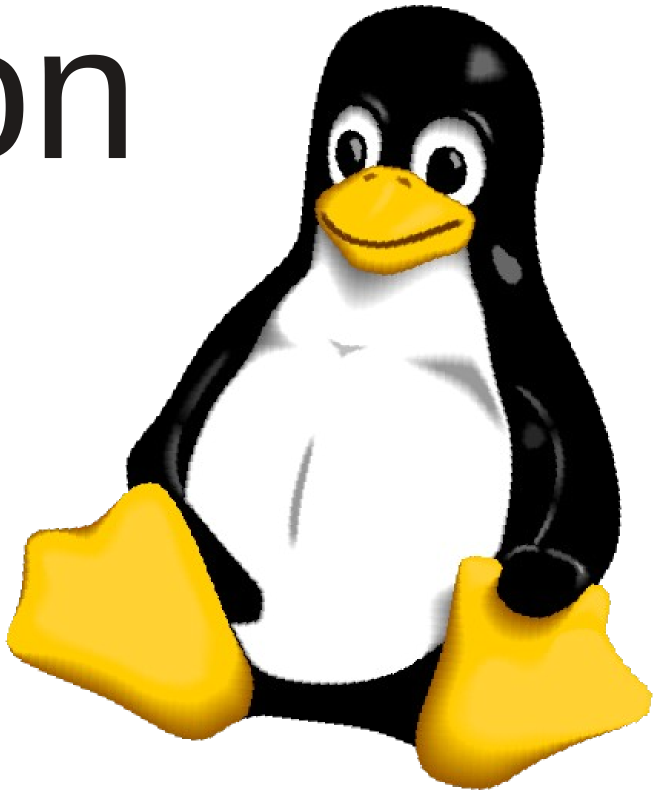
Other OS's

- Contiki OS
  - Adam Dunkels
    - Sweedish Institute for Computer Science
    - Author of **uIP** and **lwIP**
  - Supports IPv6, 802.15.4, and 6LoWPAN
  - Runs on small to tiny CPUs
    - MC1322x, AVR, 6502, others
  - Not real-time, but uses **protothreads**
  - http://www.contiki-os.org/
  - BSD License

Other OS's

- TinyOS
  - Maintained by the TinyOS Alliance
    - Started with UC Berkeley, Intel Research, and Crossbow Technologies
  - Runs on slightly larger hardware than Contiki
    - MSP430, ATmega128, XScale PXA271
  - Applications written in **nesC**, similar to C
    - Custom GNU Toolchain
  - Has support for **Beacon-Enabled** Networks

Demonstration



Embedded Linux Conference Europe

Demo

- Hardware Used
  - Node 1
    - BeagleBone
    - Microchip MRF24J40MA
    - Maxbotix Ultrasonic Range Finder (HRLV-EZ0)
  - Node 2
    - Laptop
    - Redwire Econotag

Demo

- Microchip MRF24J40MA
  - FCC, IC, ETSI certified (US, Canada, Europe)
  - Fully integrated module, only needs SPI connection
  - 2.4 GHz, 0 dBm (1 mW)
  - $10 USD for single units
  - Supported by Mainline kernel
    as of 3.7-rc1

Demo

- Redwire Econotag
  - Mariano Alvira, Redwire LLC
    - http://www.redwirellc.com/
    - http://mc1322x.devl.org/
  - Based on Freescale MC13224
    - ARM7 SOC
    - Integrated 802.15.4 radio (4.5 dBm)
    - JTAG and console over USB (FTDI)
    - Debug with OpenOCD and GDB
    - Well supported by Contiki-OS
    - Firmware to connect to the Linux 802.15.4 Serial driver.

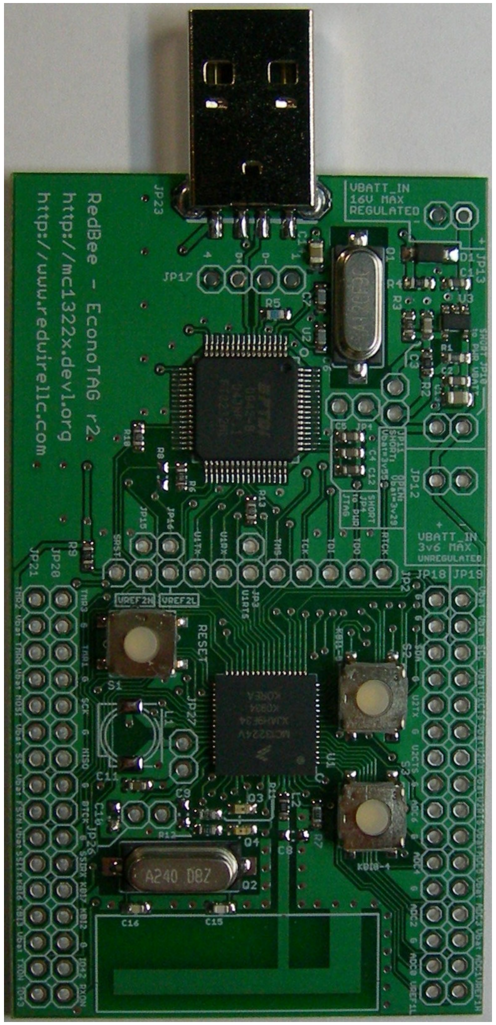


Image from Redwire, LLC

Embedded Linux Conference Europe

Demo

- BeagleBone
  - Texas Instruments / CircuitCo
  - AM3359, ARM Cortex-A8 SOC
  - 3.3v I/O, 0.1” spaced connectors
  - Boots mainline kernel +patches
  - Ethernet, USB host and device Micro SD
  - Great for breadboard prototypes
  - http://www.beagleboard.org

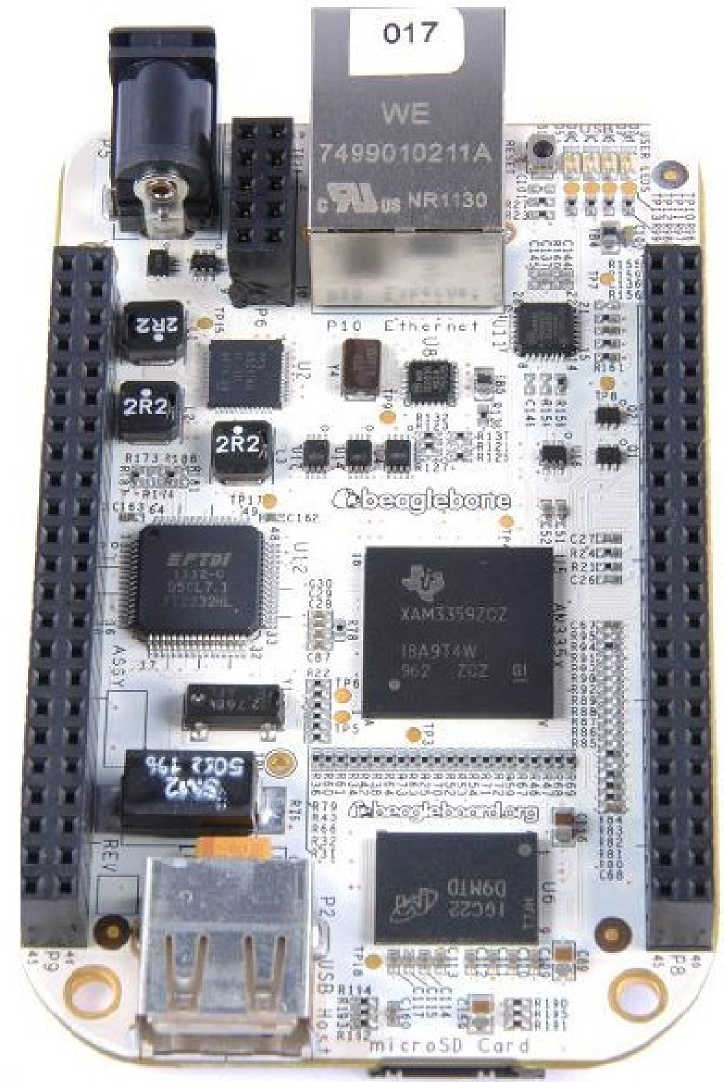


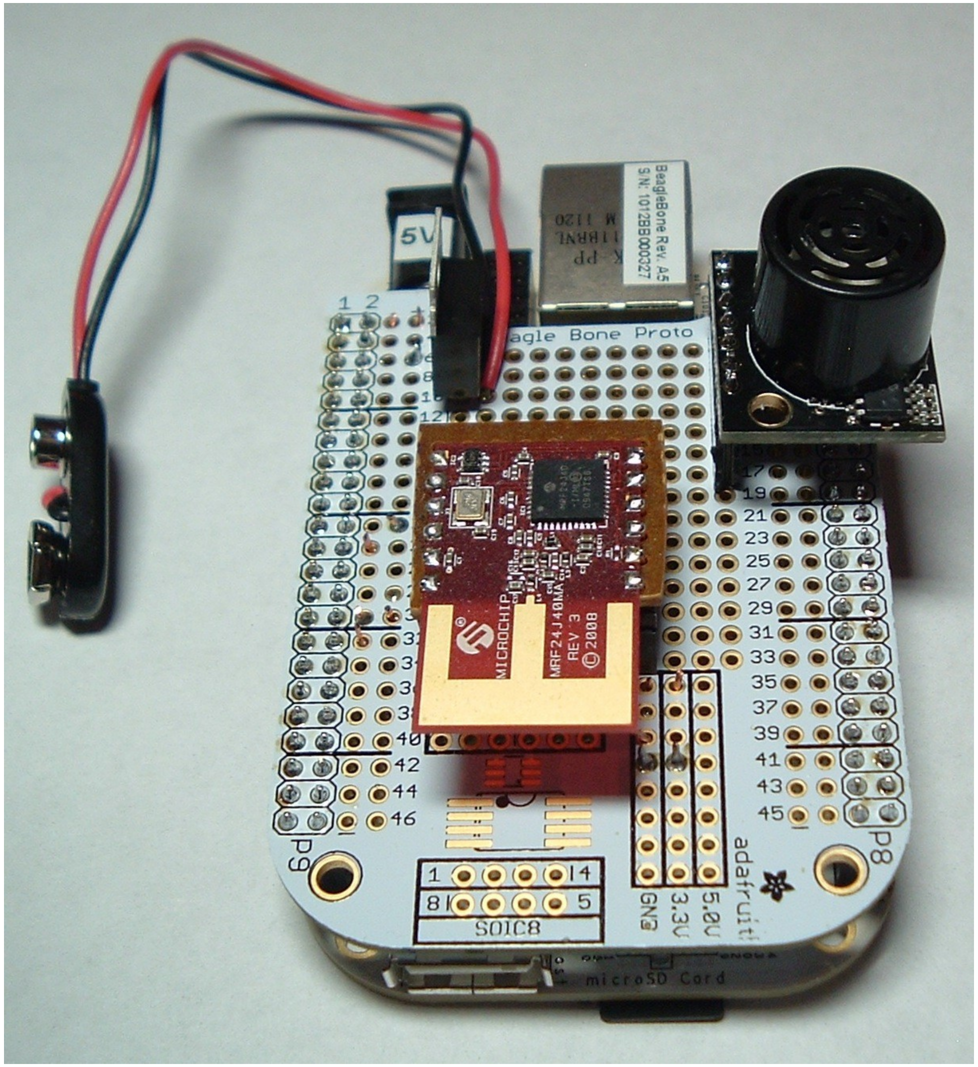
Image from Beaglebone SRM

Demo

- Application
  - Security System
    - Ultrasonic range sensor attached to the BeagleBone
      - Maxbotix HRLV-EZ0, connected to UART2
    - Alarm console on PC
      - Set, Unset, Alarm indication
    - When alarm is set, and range sensor detects a person, the alarm trips.
      - Alarm is indicated until reset
    - UDP packets send commands and indicate alarm trip.

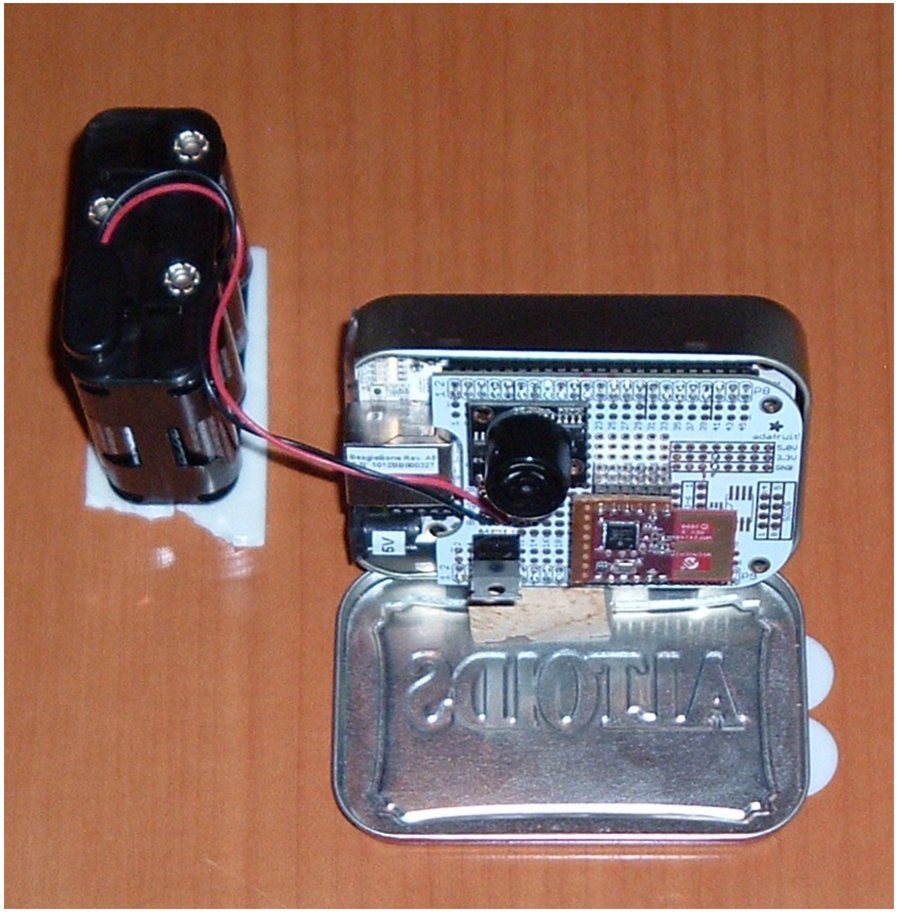
Demo

- Sensor Board
  - Adafruit Proto Cape Kit
  - Microchip MRF24J40MA
  - Maxbotix HRLV-EZ0
  - LM7805 (5V regulator)
  - Battery Snap Connector



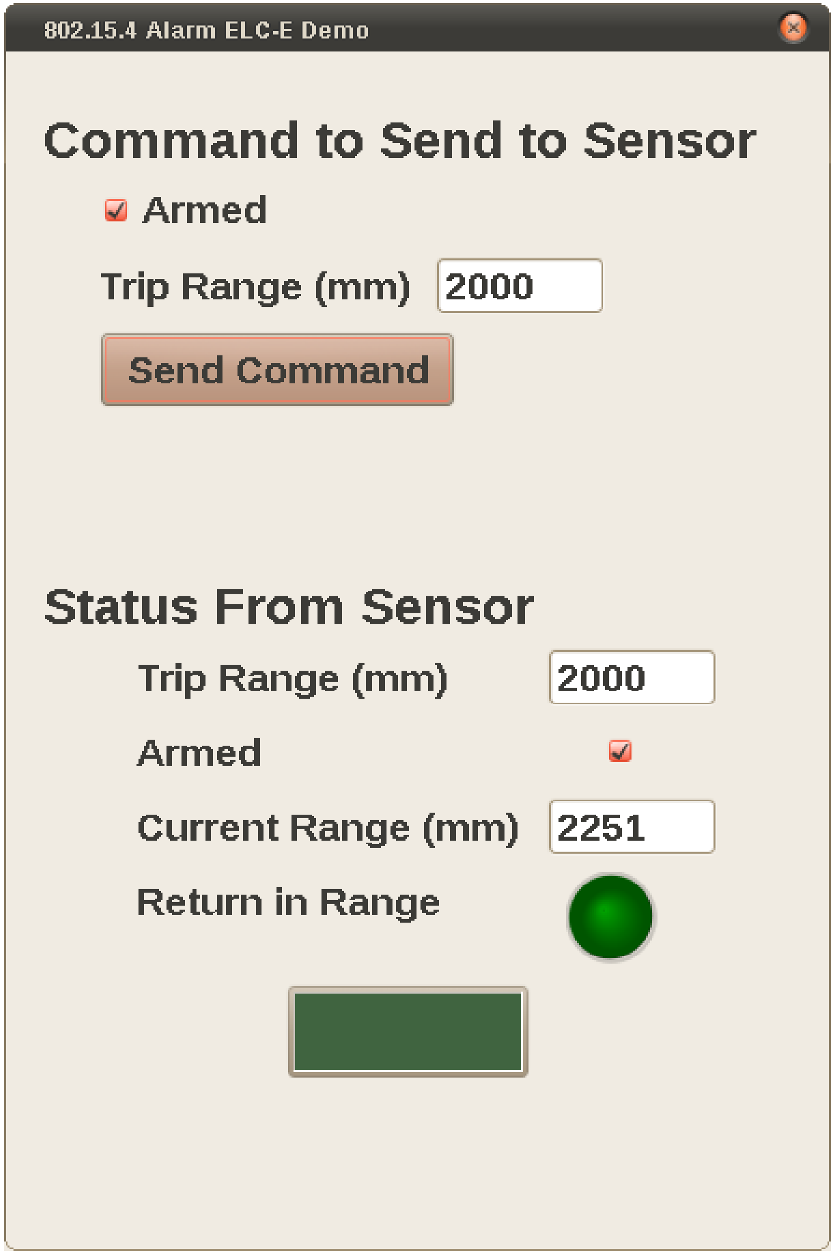
Demo

- Installed
  - Mounted in an Altoids tin over the door behind you.
  - Mounted at angle, sensor facing down.
  - A piece of cork holds the tin open at the right angle.
  - (picture is from below, looking up at it).



Demo

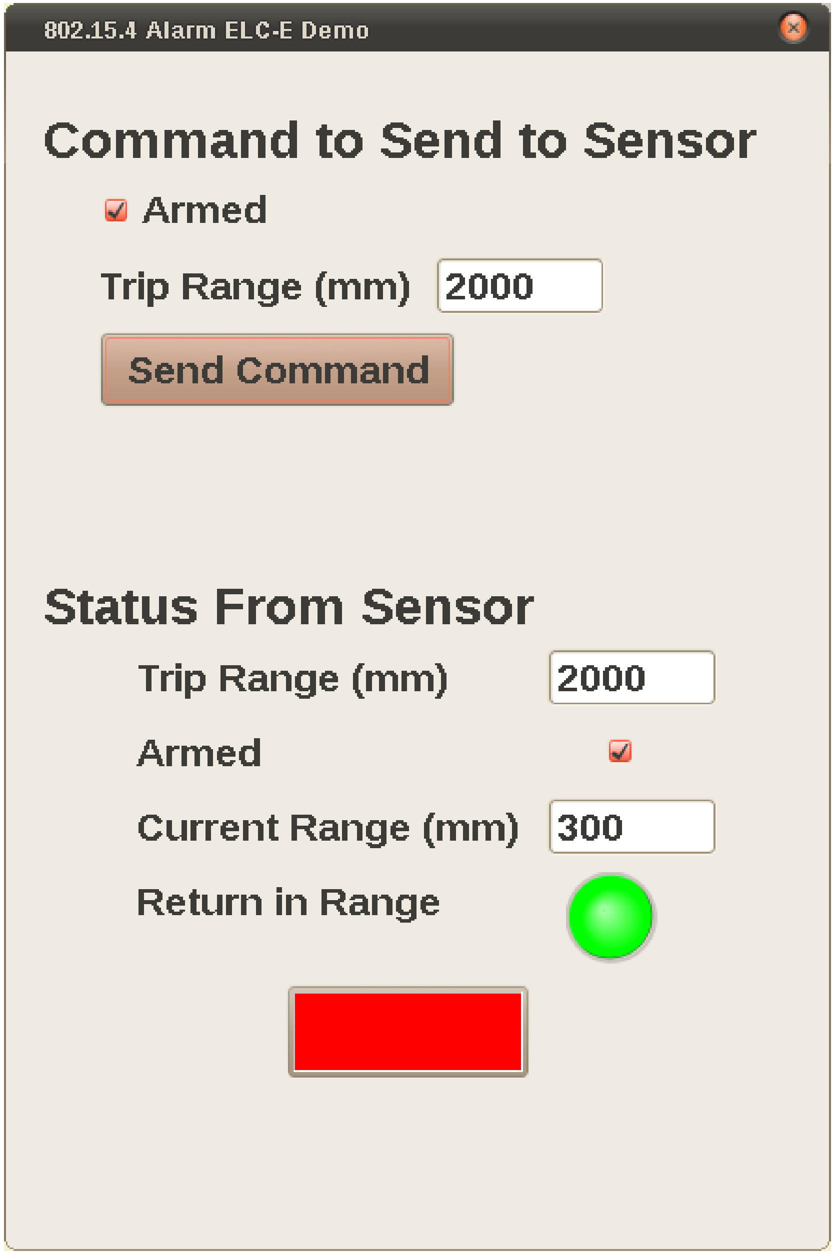
- Controller GUI
  - Alarm not tripped



Embedded Linux Conference Europe

Demo

- Controller GUI
  - Alarm tripped
  - Current return is in inside range threshold



Embedded Linux Conference Europe

Demo

- Source Code
  - Mainline Kernel 3.7.0-rc2 (PC)
  - BeagleBone kernel from:
    - https://github.com/beagleboard/kernel/tree/3.7
  - Resources downloadable from:
    - http://www.signal11.us/oss/elce2012
      - Tony Cheneau's 6lowpan and ieee802154 fixes
      - Source code for **sensor** and **controller** software
      - Kernel device tree mods for **BeagleBone**
      - Hacky board stub file (mrf24j40 driver has no DT support)
      - Hack to slow down the Econotag TX (since we don't do acks).

Lessons Learned

- 6LoWPAN requires all fragments of a single IPv6 packet to be received at the same time.
  - No re-transmission request at the 6LoWPAN layer.
  - MAC-level acknowledgement and retransmission is really needed to make it work.
- Accounting for different speeds of hardware is important.
- The mainline BeagleBone kernels are experimental!
  - It's hard to be on the bleeding edge of two things at once.

Acknowledgements

- Presentation Reviewers:
  - Tony Chenau
- #beagle (freenode) help desk:
  - Koen Kooi
  - Matt Porter
  - Hunyue Yau
  - Matt Ranostay
  - Pantelis Antoniou
- Hardware Support:
  - Aaron Wiginton

Signal11

SOFTWARE

Alan Ott

alan@signal11.us

www.signal11.us

+1 407-222-6975 (GMT -5)